4. Denise

Program Name: Denise.java **Test Input File: denise.dat**

Denise loves to go to concerts. However, every concert she goes to, she tends to drop nearly all the contents of her purse. She always realizes when it's about time to leave, but she has all her friends with her. Since it is time to go, she is in a hurry and needs your help to determine how to most quickly retrieve the contents of her purse, now scattered across the location.

Since Denise has all her friends with her, she can use as many as necessary to help her retrieve her items, but she needs to optimize how far each of her friends needs to go. Due to the crowd, oftentimes it isn't quite a straight shot between the edge of the concert grounds and her purse contents, so her friends will need to snake through the crowd. However, if on the path to retrieve another item from her purse, a friend walks past a different item, they should go ahead and pick up that item too. Therefore, you are trying to find the shortest path to each object, but, if the shortest path to an object also walks past another object, then the friend should pick up that object too.

If to grab an item two friends need to walk partially the same path and split off, that is acceptable; many of her friends can occupy the same space at the same time. The start of any of her friend's paths must be any non-obstacle space (not a #) **on the edge of the spill area**. Denise is always too traumatized and flustered when this happens, therefore never helps to pick up her own items; only her friends will pick up her spilled items.

```
####.##.##
###○....##
###...#○##
###○###○##
#O..###○##
#...###○##
#.....O..#
###.######
###..O..##
#######.##
```

```
####A##B##
###AA..B##
###A..#B##
###A###B##
#AAA###B##
#...###B##
#.....BB.#
###.######
###..CCC##
#######C##
```

The first sample data set is shown in the first box, with the solution depicted in the second box. For this situation, Denise only needs 3 friends, indicated by the letters A, B and C in this diagram, showing the path they took to retrieve the items they encountered. This is true because the shortest path to the **bold** objects provides access to all the other items.

```
##########
###....○##
###...####
###.######
#○..#....#
###.######
###.#....#
###.######
###.....##
#######.##
```

```
##########
###..BBB##
###BBB####
###B######
#AA%#....#
###%######
###%#....#
###%######
###%%%%%##
#######%##
```

In these two boxes, the situation and solution for the second data set are shown. Only two friends are needed, and as you can see, they walk the same path for a while, indicated by the % symbol starting at the bottom right of the map, finally branching off to retrieve the two items that were spilled.

Denise (cont)

Given a map of the spill area, and the location of Denise's purse contents, find the shortest path to the objects. If a shortest path to a farther item contains the most number of other items, choose that path. Determine how many such paths exist (and thus how many of Denise's friends are needed) to collect *all* her items.

Sample Input:

```
3
####.##.##
###○....##
###...#○##
###○###○##
#○..###○##
#...###○##
#.....○..#
###.######
###..○..##
#######.##
-
##########
###....○##
###...####
###.######
#○..#....#
###.######
###.#....#
###.######
###.....##
#######.##
-
##########
###.○..○##
###...####
###.######
#○..#....#
###.######
###.#....#
###.######
###....○##
#######.##
-
```

Sample Output:

```
3
2
2
```